Semantic Monitoring Techniques for EUD

José A. Macías and Pablo Castells
E.P.S. Universidad Autónoma de Madrid.
{j.macias, pablo.castells}uam.es

Abstract

In this work, we present how domain modeling and Programming by Example techniques can be combined to carry through a EUD approach. Our techniques are based on detecting iteration patterns from user monitoring as well as extracting knowledge about the user interface itself. Combining those, dynamic behavior can be characterized, getting maximum amount of semantic at each user step. This approach can be used in order to make PBE inference process more effective as well as to result in enhancing PBE global efficiency.

Keywords: End-User Development, Programming By Example, Intelligent User Interfaces, Model-Based User Interfaces, Knowledge Extraction.

1. Introduction

PBE can be considered as a main order concern in EUD paradigm [4]. The main purpose of PBE [2, 5] techniques is to help a non-expert in programming user create a programmatic representation of her actions. Such representation can be executed by the system on behalf of the end-user.

Concerning PBE, several mechanisms have been proposed in order to infer information at each user step. For instance, initial and final approach [3] tries to analyze the state of the application before and after making changes. Other mechanisms have been used in PBE, such as inspecting object's values of the application [8] in order for the system to keep track of the sequence of changes. By contrast, user monitoring techniques have been applied in some other works [5] to get a more detailed sequence of what the user has tried to do during the whole interaction.

However, most of previous related techniques are suitable for a reduced set of user interfaces. Although such techniques might be theoretically applied to all kind of user interfaces, they are mostly experimental as well as domain-dependent since a great deal of implicit information about the interface is needed for the inference process.

We believe in explicit domain (i.e. application model) information as a way to characterize user's intents. Since user's actions are mostly related to interface information (i.e. presentation and an extract of domain information), semantics about interface can be thought of as essential in supplying additional information to the inference process in order for PBE systems to better infer user's intents.

The use of domain models became an important concern in early works on Model-Based User Interfaces [7]. Data as well as presentation and behavior models are likely to be considered as a high level specification used in the automatic generation of user interfaces. Such models supply a great deal of expressivity that can be used, together with user's monitored actions, to work out what the user is doing or going to do at every step.

Assumptions made above are intended to enhance PBE at the inference process, in order for PBE systems to produce a high level of trust and be domain-independent.

2. An example

To put into practice the above ideas, we present a real scenario where the challenge of improving PBE with interface's semantics can be applied. Such an example is related to

dynamic web authoring, that is to say how a non-expert user can modify dynamically generated web pages by herself.

The dynamic generation of web pages has become commonplace for even the simplest WWW applications today. However, authoring dynamic web pages is not an easy task. Most of these pages are generated by means of web programming languages so that non-expert in programming users cannot afford to face editing these pages.

Here we have a real scenario where PBE is required in order to supply the non-expert user with an authoring tool that enables her to modify the dynamic document. This situation can be conceptually easy to understand but is very hard to overcome. While most existing authoring tools are suitable for static web pages, we are not aware of any WYSIWYG tool that can automatically generate the page generation procedure.

PBE techniques might help met some of the requirements for dynamic web authoring, since inference process can automatically produces the generation procedure of the dynamic web page. However, in this case, the use of domain knowledge will be useful in order to obtain a more detailed representation of the generation procedure.

For this scenario we propose a WYSIWYG interface where the user can freely modify the page. Therefore the authoring tool infers a programmatic representation of changes, modifying the generation procedure that will affect the future generation of other pages. Furthermore, user monitoring can be applied, as well as the information domain, to get a high level of reliability to the inference process. This will produce a more suitable inference, detecting automatically relationships between user's actions and interface widgets (here HTML tags). In that example, the use of domain model might help find out more complex user's actions, such as automatic transformation between presentation widgets. For instance, the system can build a whole table coming from a reduced set of user's actions as long as the system can carry out mappings between the table and the domain knowledge information.

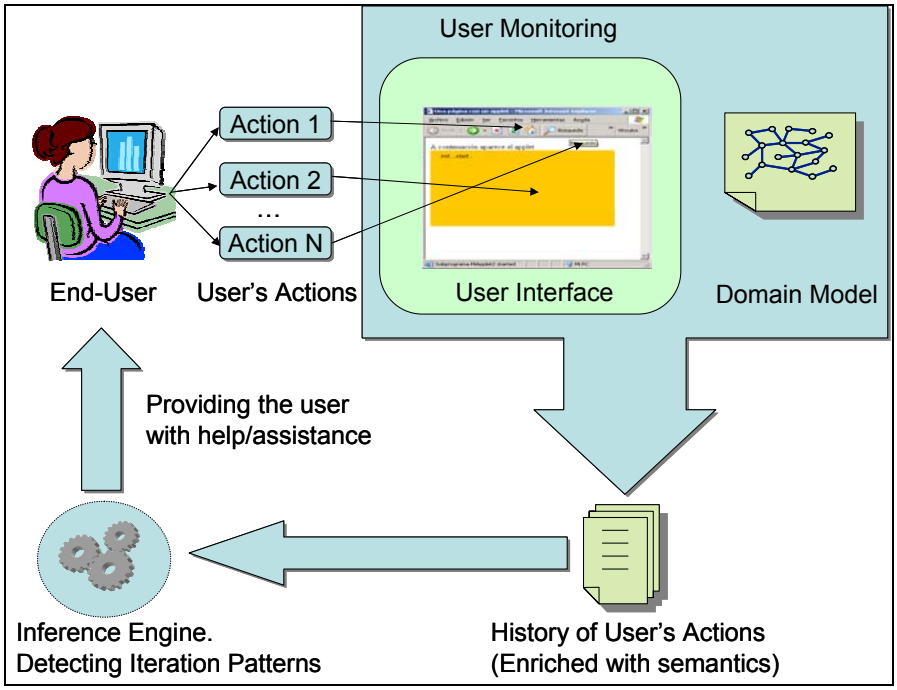

Figure 1. Our PBE add-on approach

3. Detecting Iteration Patterns

One of the advantages in using a monitoring model is that a semantic history of user's actions can be built in real time. In our approach the system analyzes and manages such history to find iteration patterns from user's actions.

Iteration patterns can be though of as a generalization of common user's actions that can appear more than once, thus they can be used to apply similar behavior on future interactions. This way, the authoring tool can record and recognize iteration patters to suggest the user to achieve cumbersome tasks on her behalf.

In our approach (Figure 1), the system tracks user's actions and then uses domain information to construct a semantic history of user's actions. Such semantics contain interface's components references as well as internal annotations about generation. Furthermore the system also detects and models presentation structures like tables and selection lists. Later, an inference engine processes the historic information to detect iteration patterns than can be applied to assist the user on the interaction. This way the system is able to provide the end-user with help as well as to perform task on behalf of her

4. Experience

Our experience is in PBE systems that help user editing dynamic web pages. Actually we have developed an authoring tool, namely DESK [6] that assists the user to carry out changes to dynamic web pages. DESK infers user's intents by monitoring user activity and extracts meaningful information from the web page that is used to get semantic relationships between user's actions and interface information (i.e. HTML page).

DESK is intended to reduce the gentle slope of complexity, supplying with a WYSIWYG easy-to-use user interface but, in contrast, featuring some expressive limitations since DESK is focused on WYSIWYG representations rather than abstract ones.

DESK has been used in several scenarios, and an empirical evaluation has been achieved in order to evaluate the usability of the authoring tool taking into account the user's opinion.

Furthermore, we have previous experience in developing PBE systems such as HandsOn [1] that uses explicit information from a data model for detecting relationships between visual and textual components.

ACKNOWLEDGEMENTS

The work reported in this paper is being supported by the Spanish Ministry of Science and Technology (MCyT), project number TIC2002-1948.

REFERENCES

1. Castells, P.; Szekely, P.: Presentations Models by Example. In Design, Specifications and Verifications of Interactive Systems'99. D.J. Duke and A. Puerta (eds.). Springer-Verlag, Viena, 1999, pp. 100-116.
2. Cypher A. (ed.).: Watch What I Do: Programming by Demonstration. The MIT Press, 1993.
3. Frank, M.; Foley, J.: A pure reasoning engine for programming by demonstration. Proceedings of the ACM Symposium on User Interface Software and Technology. Marina del Rey, CA, November 2-4, 1994, pp. 95-101.
4. Klann, M.; Fit, F.: End-User Development. D1.1 Roadmap. Proceedings of the End User Development Workshop at CHI'2003 Conference. Ft. Lauderdale, Florida, USA. April 5-10, 2003.
5. Lieberman, H. (ed): Your Wish is my Command. Programming By Example. Morgan Kaufmann Publishers. Academic Press, USA. 2001.
6. Macías, J.A.; Castells, P.: Dynamic Web Page Authoring by Example Using Ontology-Based Domain Knowledge. Proceedings of the International Conference on Intelligent User Interfaces (IUI'03). Miami, Florida, USA. January 12-15, 2003, pp. 133-140.
7. Macias, J.A.; Castells, P.: Using Domain Models for Data Characterization in PBE. Proceedings of the End User Development Workshop at CHI'2003 Conference. Ft. Lauderdale, Florida, USA. April 5-10, 2003.
8. McDaniel, R.G.; Myers, B.A.: Getting More Out Of Programming-By-Demonstration. Proceedings of the CHI'99 Conference. Pittsburg, PA, USA, 1999.